VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

ÚSTAV TELEINFORMATIKY

Ing. Václav Pfeifer

DETEKCE KLÍČOVÝCH SLOV V ŘEČOVÝCH SIGNÁLECH

KEYWORD DETECTION IN SPEECH DATA

ZKRÁCENÁ VERZE PH.D. THESIS

Obor: Teleinformatika

Školitel: Ing. Miroslav Balík, Ph.D.

Oponenti:

Datum obhajoby:

KEYWORDS
classifier, frame-based, phoneme, detection, hierarchical, speech

KLÍČOVÁ SLOVA
klasifikator, ramcovy, fonem, detekčni, hierarchicky, řeč

Disertační práce je k dispozici na Vědeckém oddělení děkanátu FEKT VUT v Brně, Technická 3058/10, 616 00 Brno

OBSAH

1 ÚVOD

Řeč je v současnosti nejčastějším způsobem komunikace mezi lidmi. Proces vytváření řeči v lidském těle je založen na skupině orgánů obecně nazývaných jako řečové orgány. Tyto řečové orgány tvoří dohromady hlasový trakt, který se dá rozdělit na tři základní ústrojí – dechové, hlasové a artikulační. Výsledkem je pak produkce základních řečových jednotek – fonémů a alofonů [22]. Posloupnost těchto jednotek postupně tvoří nadřazené jednotky – slova. Zřetězením jednotlivých slov vyjadřujeme základní myšlenku(y) (informaci, sémantiku). Na rozdíl od textového způsobu komunikace jsou informace obsažené v řeči obohaceny o spoustu dalších informací (identita řečníka, aktuální nálada, aj.). Tyto aditivní informace komplikují práci s řečovými signály.

Číslicové zpracování řečových signálů zaznamenalo v poslední době značný rozmach s příchodem výkonných výpočetních systémů. První pokusy o vytvoření systému, který by byl schopen detekovat klíčová slova v řečových signálech, byly uskutečněny již v 80. letech minulého století. Tyto systémy byly založeny převážně na principu porovnávání vzorů TM (Template Matching) a kvůli velmi omezené výpočetní schopnosti tehdejších systému byly možnosti těchto systémů velmi limitované. Začátkem 90. let, s příchodem výkonných výpočetních systémů, byly vytvořeny první systémy, které již byly s úspěchem aplikovány v praxi. Bohužel, tyto systémy byly značně ovlivněny aktuálním nastavením daného řečníka (závislé na určitém řečníkovi), a proto byla praktická aplikace převážně v systémech pro hlasové ovládání (telefon, hlasová verifikace, atp.). Problémy se závislostí na řečníkovi byly do značné míry vyřešeny až s příchodem moderních statistických metod pro modelování klíčového slova spolu s metodami pro normalizaci hlasového traktu VTLN (Vocal Tract Length Normalization), MLLR (Maximum Likelihood Linear Regression) aj. [9].

Detektor klíčových slov je komplexní systém, který v sobě zahrnuje subsystémy pro extrakci příznaků, klasifikaci, normalizaci, atd. a každý z těchto subsystémů zásadně ovlivňuje výsledky detektoru. Současné metody pro extrakci příznaků jsou převážně založeny na kepstrální analýze vstupního signálu (Mel-frekvenční kepstrální koeficienty MELFCC (Mel-Frequency Cepstral Coefficients) [22], vjemové koeficienty PLP (Perceptual Linear Prediction) [22]), nebo na komplexních systémech, založených na extrakci parametrů v delším časovém měřítku (Systémy založené na TANDEM architektuře [22, 12]).

Stěžejním součástí každého detekčního systému je klasifikátor. V případě detektoru klíčových slov se obvykle jedná o fonémový klasifikátor. Většina současných klasifikátorů pro detekci klíčových slov je založená na popisu pomocí směsi Gaussových křivek GMM (Gaussian Mixture Models) nebo na neuronových sítích

NN (Neural Networks). Hlavním důvodem pro aplikaci výše uvedených klasifikátorů je relativně jednoduchá implementace a dobré výsledky.

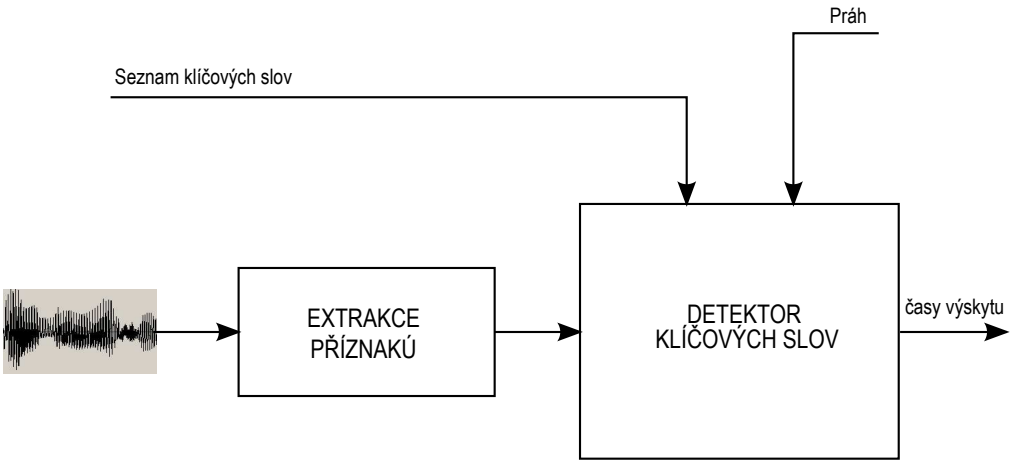
Klasifikátory založené na tzv. jádrových funkcích (Kernel Methods) existují již delší dobu, ale až s příchodem metod podpůrných vektorů SVM (Support Vector Machines) se začalo s praktickou aplikací do systémů pro zpracování řeči. Potenciál těchto metod je veliký a dosavadní výsledky to jasně dokazují [3].

Velká část současných systémů pro detekci klíčových slov je založena na robustních statistických modelech a stavových automatech. Tyto systémy se vyznačují vysokou přesností a spolehlivostí, ale i přes velké úsilí mnoha vědců jsou limitovány použitými technologiemi. Z pohledu uživatele lze detektory klíčových slov považovat za slovní klasifikátory, a proto můžou být s úspěchem aplikovány metody založené na SVM.

2 SOUČASNÝ STAV

Detekce klíčových slov je obor, který se snaží vyhledávat konkrétní informace v obecně neomezených řečových záznamech. V současné době je většina technik zajišťujících tento cíl založena na třech hlavních přístupech:

- Akustické modelování klíčového slova
- Aplikace rozpoznavače řeči s velkým slovníkem LVCSR (Large Vocabulary Continuous Speech Recognizer)
- Metody založené na porovnávání vzorů TM (Template Matching)



Obr. 2.1: Obecné schéma detektoru klíčových slov

Blokové schéma obecného detektoru klíčových slov KWS je na obr. 2.1. Navzorkovaný řečový signál je transformován na posloupnost koeficientů reprezentujících vstupní řečový signál – vektor příznaků. Příznaky jsou ještě obvykle normalizovány a následně vstupují do bloku detektoru, společně se seznamem klíčových slov a prahovacími úrovněmi. Výstupem detektoru pak může být:

– identita klíčového slova – „co“
– časy výskytu – „kde“
– jistota (konfidence) klíčového slova – „jak si je systém jistý“

Princip akustických metod je právě v akustickém modelování klíčového slova společně s výplňovým (někdy také nazývaný „garbage“) modelem. V obou případech se využívá Markovových modelů HMM (Hidden Markov Models) společně s jedním z následujících klasifikátorů:

- Směsi Gaussových křivek GMM
- Neuronových sítí NN
- Metody podpůrných vektorů SVM

HMM a klasifikační metody jsou v systémech pro zpracování řeči úzce propojeny, a proto se tyto metody označují souhrnně jako HMM/GMM, resp. HMM/NN. Ačkoliv jsou v technické praxi systémy založené na HMM/GMM a HMM/NN dominantní, s úspěchem byly prezentované metody založené na aplikaci nelineárních jádrových (kernel) funkcí. Tyto metody vycházejí z principu SVM a jsou prezentované např. v literatuře [15, 3]. Na základě určité rozhodovací úrovně (prahu – treshold) může být výstupem detektoru posloupnost časů výskytu zadaných klíčových slov, nebo jen obecné rozhodnutí, zda se klíčové slovo nachází v řečovém signálu. Většina detektorů navíc poskytuje informaci o věrohodnosti výskytu klíčového slova.

Detektory využívající rozpoznávač řeči (obvykle s velkým slovníkem – LVCSR) pracují ve dvou krocích. V prvním kroku je využitím LVCSR rozpoznávače provedena textová transkripce. Detekce klíčového slova je zjednodušena na prohledávání textového prostoru. Výhodou LVCSR systémů je, že výstupem nemusí být vždy nejlepší posloupnost rozpoznaných slov, ale také strom hypotéz (lattice), se kterým většina těchto systémů také pracuje. Nevýhodou je závislost na vstupním slovníku – v případě, že se v záznamu vyskytuje výraz, který není obsažen v rozpoznávacím slovníku, nedojde k rozpoznání a v textovém přepisu se tedy slovo neobjeví. Tuto situaci obecně nazýváme jako detekci mimo slovník OOV (Out Of Vocabulary) [23, 13].

Jak akustická metoda, tak metoda LVCSR se řadí do kategorie statistických metod. Princip těchto metod spočívá v modelování základních jednotek (slov, fonémů) obecným (generalizovaným) modelem. Nejčastěji je využíváno právě kombinace následujících modelů – HMM/GMM a HMM/NN. Výhodou obecných modelů je relativně nízký počet parametrů popisujících daný model (střední hodnota $\boldsymbol{\mu}$, kovarianční matice $\boldsymbol{\sum}$, vektor vah $\mathbf{w}$ a matice přechodu $\mathbf{a}$) a implementačně jednoduché algoritmy pro odhad těchto parametrů. Nejpopulárnější je trénovací postup založený na kritériu maximální věrohodnosti ML (Maximum Likelihood) – např. metoda BW (Baum-Welsch), která je založena na populární metodě očekávané maximalizace EM (Expected Maximalization). Princip metody byl již mnohokrát popisován a lze najít např. v [4, 1]. Výhodou trénovacího algoritmu je, že v každém kroku, kdy probíhá odhad nových parametrů, zaručuje zlepšení odhadu. Nevýhodou algoritmu je častá konvergence k lokálnímu maximu. Z těchto důvodů se často využívají alternativní přístupy trénování – tzv. diskriminativní trénování. Jedním z nejznámějších přístupů jsou metody:

- MCE (Minimum Classification Error) [15]
- MMI (Maximal Mutual Information) [18]

Z historického hlediska byly první detektory založeny na porovnávání vzorů – TM. Principem je vzájemné srovnání testovacího a referenčního slova (na akustické úrovni). Nejznámější je metoda dynamického borcení času DTW (Dynamic

Time Warping) založená na nelineárním časovém zarovnání referenčního slova a následném vyhodnocení. Hlavními nevýhodami TM je velká závislost na konkrétním řečníkovi a také fakt, že současné algoritmy nedokáží efektivně využívat trénovací data ze současných řečových korpusů (na rozdíl od fonetického modelování u HMM) [11]. Další nevýhodou je rychlost algoritmů v případě velkého vstupního slovníku společně s více referencemi pro každé klíčové slovo. Byly navrženy různé modifikace – např. autoři v literatuře [2] navrhují aplikaci nových příznaků (posteriors) a zároveň nahrazují standardní eukleidovskou vzdálenost (L2 normu) Kullback-Leibler (KL) vzdáleností [2]. Změnu metriky pro výpočet L2 normy navrhují také autoři v [10]. Principem je aplikace Neuronové sítě a zavedení nové metriky pro výpočet mezi-rámcových vzdáleností [10, 6].

Ačkoliv autoři, zabývající se metodami založenými na TM, mnohdy prezentují lepší výsledky, než je tomu u HMM přístupů, je praktická aplikace TM algoritmů nerealizovatelná – dekodér LVCSR systému založeném na TM (s osmi vzory na slovo) potřebuje řádově mnohem vyšší dobu než LVCSR založený na HMM. Zajímavým přístupem, který principiálně vychází z TM přístupu, je aplikace nelineárních funkcí na daný řečový signál DKWS (Discriminative KeyWord Spotting). Princip metody vychází z „Large margin and kernel“ metod, které jsou založeny na transformaci příznakového vektoru do vysoce rozměrného vektorového prostoru se skalárním součinem (v obecném případě se pracuje s Hilbertovým prostorem $\mathcal{H}$) s nelineární bází. V tomto novém prostoru se již dá sestrojit rozhodovací nadrovina, která separuje jednotlivé třídy (slova, fonémy, aj.) [3]. Autoři v literatuře [14, 10, 16, 17, 15] aplikují sadu nelineárních příznakových (feature) funkcí, kde každá funkce představuje jistotu výskytu jednotlivého fonému (resp. klíčového slova) v zadaném čase a v zadaném úseku rámce řečového signálu (L2 norma, fonémový klasifikátor [8], průměrná délka fonému, aj.). Každá příznaková funkce je ohodnocena určitou vahou a úkolem trénovacího procesu je určit tyto jednotlivé váhy.

Velkou výhodou SVM přístupu je možnost definice chybové funkce detektoru a její následná minimalizace (např. pomocí metody SGD (Stochastic Gradient Descent) [3, 15]). Vzhledem k tomu, že definovaná chybová funkce je konvexní, zaručuje algoritmus konvergenci ke globálnímu optimu (narozdíl od trénovacího algoritmu BW u HMM) [15]. Další výhodou je efektivní využití současných řečových korpusů. DKWS pracuje primárně s akustickým modelem, takže odpadají problémy s trénováním jazykového modelu.

3 LINEÁRNÍ KLASIFIKÁTORY

3.1 Lineární hierarchické klasifikátory

3.1.1 Definice problému

Nechť $\bar{\mathbf{x}}$ reprezentuje posloupnost akustických příznakových vektorů takových, že $\bar{\mathbf{x}} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T), \mathbf{x}_t \in \mathcal{X}$ pro každé $1 \leq t \leq T$, kde $\mathcal{X} \subset \mathbb{R}^n$ představuje doménu všech akustických příznakových vektorů. Nechť $\mathcal{Y}$ je množina všech fonémů a fonémových skupin definovaných dle hierarchické fonetické struktury (viz obr. 3.1). V hierarchické klasifikaci $\mathcal{Y}$ má dvě hlavní funkce – jako u většiny více-třídových klasifikačních úloh přiřazuje ke každému fonému $v \in \mathcal{Y}$ množinu všech odpovídajících akustických příznakových vektorů $\mathbf{x} \in \mathcal{X}$ a současně definuje množinu všech vrcholů (fonémů a fonémových skupin) v hierarchické stromové struktuře [8, 20]. Množina všech fonémů a fonémových skupin je značena jako $k = |\mathcal{Y}|$ a je předpokládáno, že $\mathcal{Y} = \{0, \ldots, k-1\}$, kde 0 představuje uzel (root) stromové struktury $\mathcal{T}$.

Pro libovolnou dvojici fonémů $u, v \in \mathcal{Y}$ definujeme jejich vzdálenost v stromové struktuře jako $\gamma(u, v)$. Tato vzdálenost $\gamma(\cdot, \cdot)$ definuje jednoduchou metriku, která reprezentuje počet hran mezi fonémy u a v ve stromové struktuře $\mathcal{T}$. Vzhledem k tomu, že funkce $\gamma(\cdot, \cdot)$ je nezáporná, splňují následující vztahy $\gamma(u, v) = \gamma(v, u)$ a $\gamma(u, u) = 0$ trojúhelníkovou nerovnost [8]. Definujme dále tzv. stromově indukovanou chybu $\gamma(u, v)$ jako nejmenší počet hran mezi fonémy u a v ve stromové struktuře $\mathcal{T}$. Toto tvrzení implikuje fakt, že chyba nastává pouze při nesprávné fonémové klasifikaci.

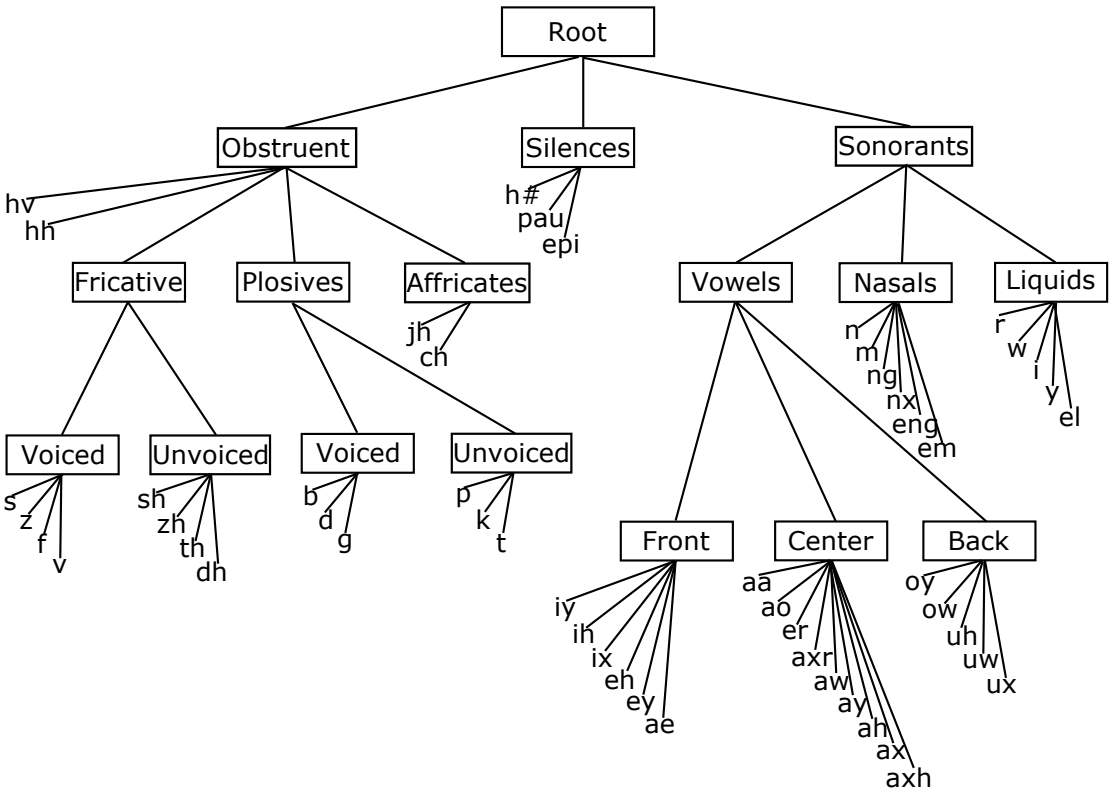
Pro každý foném a fonémovou skupinu (vyjímaje kořenového uzlu 0) $v \in \{\mathcal{Y} \backslash 0\}$ je definován jejich předchůdce (rodič) ve stromové struktuře $\mathcal{Y}$ jako $\mathcal{A}(v)$. Dále je pak rekurzivně definován i-tý předchůdce jako

$$\mathcal{A}^{(i)}(v) = \mathcal{A}(\mathcal{A}^{(i-1)}(v)) \tag{3.1}$$

a současně platí, že $\mathcal{A}^{(0)}(v) = v$. Jinými slovy, $\mathcal{A}(v)$ je přilehlý vrchol k fonému v, který má ve fonetické struktuře kratší vzdálenost ke kořenovému uzlu 0 [8]. Pro každý foném a fonémovou skupinu $v \in \mathcal{Y}$ je definována metrika $\mathcal{P}(v)$ jako

$$\mathcal{P}(v) = \left\{ u \in \mathcal{Y} \ : \ \exists i \ u = \mathcal{A}^{(i)}(v) \right\}. \tag{3.2}$$

$\mathcal{P}(v)$ reprezentuje počet fonémů a fonémových skupin (resp. počet jednotlivých vrcholů) ve stromové struktuře $\mathcal{T}$ po cestě od fonému v ke kořenovému uzlu 0. Např. zvučné „s“ má vzdálenost $\mathcal{P}(v) = 4$, protože minimální počet hran mezi fonémem „s“ a kořenovým uzlem 0 je roven čtyřem.



Obr. 3.1: Hierarchická fonémová struktura pro anglickou fonetickou abecedu

3.1.2 Klasifikační funkce

Klasifikátor (resp. klasifikační funkce) $f : \mathcal{X} \to \mathcal{Y}$ provádí predikci v závislosti na množině prototypů (váhovacích vektorů) $\mathbf{W}$ definovaných pro každý foném a fonémovou skupinu. Každý prototyp $\mathbf{W}$ je libovolný vektor definovaný v prostoru reálných čísel $\mathbb{R}^n$ a úkolem trénovacího algoritmu je nalézt takovou množinu váhovacích vektorů $\mathbf{W}^1, \ldots, \mathbf{W}^{k-1}$, která bude dosahovat minimální stromově indukovanou chybu $\gamma(\cdot, \cdot)$ na trénovacích vzorcích. Trénovací databáze $\mathcal{S}$ pro rámcový klasifikátor je reprezentována množinou uspořádaných dvojic $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$ To znamená, že množina $\mathcal{S}$ obsahuje m uspořádaných dvojic $\mathbf{x}_i \in \mathcal{X}$ a $y_i \in \mathcal{Y}$ – trénování je prováděno po dílčích rámcích. Lineární rámcový klasifikátor f je definován v následujícím tvaru:

$$f(\mathbf{x}) = \arg\max_{v \in \mathcal{Y}} \mathbf{W}^v \cdot \mathbf{x}. \tag{3.3}$$

Definice lineárního klasifikátoru podle rovnice (3.3) nezahrnuje hierarchickou strukturu. Implementace hierarchické struktury do klasifikační funkce (3.3) je provedena následovně. Jednotlivé váhovací vektory $\mathbf{W}$ jsou přepsány do tvaru:

$$\mathbf{w}^v = \mathbf{W}^v - \mathbf{W}^{A^1(v)} \tag{3.4}$$

Namísto přímé práce s dílčími váhovacími vektory $\mathbf{W}$ pracujeme raději s dílčími diferencemi mezi jednotlivými fonémy a odpovídajícími předchůdci. Původní váhovací vektor $\mathbf{W}$ lze pak na základě rovnice (3.4) přepsat do následujícího tvaru:

$$\mathbf{W}^v = \sum_{u \in \mathcal{P}(v)} \mathbf{w}^u \tag{3.5}$$

Na základě rovnic (3.5) a (3.3) lze výsledný hierarchický klasifikátor vyjádřit ve tvaru rovnice (3.6) [8].

$$f(\mathbf{x}) = \arg\max_{v \in \mathcal{Y}} \sum_{u \in \mathcal{P}(v)} \mathbf{w}^u \cdot \mathbf{x}, \tag{3.6}$$

kde $\mathbf{w}^u$ je váha fonému (resp. fonémové skupiny) u, $\mathbf{x}$ je aktuální rámec a výraz $v \in \mathcal{Y}$ definuje množinu složenou z fonému v a všech jeho předchůdců $\mathcal{A}^i(v)$.

3.2 Návrh efektivního trénovacího algoritmu pro lineární hierarchický klasifikátor

Můj navržený algoritmus je založen na klasifikační funkci definovanéo rovnicí (3.6). Princip odhadu dílčích váhovacích vektorů $\mathbf{W}$, resp. $\mathbf{w}$ je založen na iteračním procesu, kdy v každém kroku je na základě určité predikované chyby provedena patřičná změna pro odpovídající váhovací vektor $\mathbf{w}^v$ fonému v. Tato predikce je provedena vzhledem k rámcové klasifikační funkci (3.6), resp. (3.3). Algoritmus tedy predikuje nové váhovací vektory $\mathbf{w}^v$ v závislosti na dílčích rámcích $\mathbf{x}^v$ (proto rámcový klasifikátor).

Ve většině případů je k dispozici celá trénovací databáze ve formě $\mathcal{S} = \{(\bar{\mathbf{x}}_i, y_i)\}_{i=1}^m$, kde $\bar{\mathbf{x}} = \{\mathbf{x}_1, \ldots, \mathbf{x}_j\}$ reprezentuje posloupnost příznaků, které odpovídají fonému y_i. Proměnná m tedy představuje celkový počet fonémů v trénovací množině $\mathcal{S}$.

Myšlenka algoritmu založena na efektivním využití posloupnosti dílčích příznakových vektorů odpovídajících danému fonému v tak, že změna dílčích váhovacích vektorů je provedena jen jednou pro každý foném y_i z trénovací množiny $\mathcal{S}$. Výhodou tohoto přístupu je velmi dobrá generalizace a současně rychlost navrhovaného algoritmu, která je v porovnání se standardním rámcovým trénovacím algoritmem (navrhovaným autory v [8, 14]) lineární vzhledem k počtu trénovacích vzorků v trénovací množině $\mathcal{S}$ [20]. Nakonec je nutné podotknout, že váhovací vektory $\mathbf{w}^v$ odvozené navrhovaným efektivním algoritmem, dosahují srovnatelné přesnosti na evaluačních datech (více viz kapitola **??**) [19].

Základní princip mého navrženého algoritmu spočívá v modifikaci klasifikační funkce f. Jak již bylo řečeno, klasifikační funkce definovaná rovnicí (3.3), resp. rovnicí (3.6) je navržena pro klasifikaci dílčích rámců $\mathbf{x}_i$. Pro implementaci sekvenčního (dávkového) trénovacího algoritmu je nutné přepsat klasifikační funkci do následujícího tvaru:

$$f(\bar{\mathbf{x}}) = \arg\max_{v \in \mathcal{Y}} \sum_{u \in \mathcal{P}(v)} \text{mean}(\mathbf{w}_i^u \cdot \bar{\mathbf{x}}), \tag{3.7}$$

kde operátor mean představuje průměr dílčích hodnot a $\mathbf{w}_i$ reprezentuje váhovací vektor v i-tém kroku iterace. Z teorie počítačového učení, založeném na jádrových (kernell) technikách a metodách maximálních hranic (Large Margin and Kernel Methods) [3], z kterých dále pak principiálně vychází techniky založené na SVM, předpokládáme, že existuje množina váhovacích vektorů $\{\mathbf{w}^v\}_{v \in \mathcal{Y}}$ takových, že pro každý trénovací pár $(\bar{\mathbf{x}}, y_i)$ a každé $r \neq y_i$ platí následující nerovnost [20, 19]:

$$\sum_{v \in \mathcal{P}(y_i)} \|(\mathbf{w}_i^v \cdot \bar{\mathbf{x}})\| - \sum_{u \in \mathcal{P}(r)} \|(\mathbf{w}_i^u \cdot \bar{\mathbf{x}})\| \geq \sqrt{\gamma(y_i, r)}, \tag{3.8}$$

kde y_i je odpovídá správné predikce podle rovnice (3.7) a $\|\cdot\|$ je $L2$ norma. Na základě rovnice (3.8) je předpokládáno, že rozdíl mezi správně klasifikovaným fonémem $v \in \mathcal{Y}$ a libovolným jiným fonémem $r \neq y_i$, je minimálně druhá odmocnina jejich vzájemné stromově indukované chyby [14]. Cílem trénovacího algoritmu je tedy nalezení takové množiny váhovacích vektorů $\{\mathbf{w}^v\}$, která bude pro splňovat nerovnici (3.8) pro všechny fonémy v. V technické praxi je obvykle nemožné splnění rovnice (3.8) pro všechny případy, a proto je v teorii počítačového učení obvyklé provádět minimalizaci nepřímo – zavedením konvexní ztrátové funkce $\ell(\{w_i^v\}, x_i, y_i)$ [3, 7].

$$\ell = \left[\sum_{v \in \mathcal{P}(\widehat{y}_i)} \|(\mathbf{w}_i^v \cdot \mathbf{x})\| - \sum_{v \in \mathcal{P}(y_i)} \|(\mathbf{w}_i^v \cdot \mathbf{x})\| + \sqrt{\gamma(y_i, \widehat{y}_i))} \right]_+ \tag{3.9}$$

Kde funkce $[z]_+ = \max\{z, 0\}$. Na základě předpokladu, že v i-tém kroku iterace byla provedena chyba nesprávné predikce $\hat{y}$, je cílem upravit všechny odpovídající váhovací vektory $\mathbf{w}_i^v$ tak, aby byly splněny podmínky definované rovnicí (3.8). Bohužel přímé řešení neexistuje, a proto je nutné převést tento problém na jednoduchou optimalizační úlohu [17]. Na základě teorie SVM je optimalizační problém přepsán do podoby následující minimalizační úlohy, která je formálně definovaná podle následující rovnice:

$$\begin{gathered} \min_{\{\mathbf{w}^v\}} \tfrac{1}{2} \textstyle\sum_{v \in \mathcal{Y}} \|\mathbf{w}^v - \mathbf{w}_i^v\|^2 \\ s.t. \textstyle\sum_{v \in \mathcal{P}(y_i)} \|(\mathbf{w}_i^v \cdot \bar{\mathbf{x}})\| - \sum_{u \in \mathcal{P}(r)} \|(\mathbf{w}_i^u \cdot \bar{\mathbf{x}})\| \geq \sqrt{\gamma(y_i, r)} \end{gathered} \tag{3.10}$$

INICIALIZACE: $\forall v \in \mathcal{Y} : \mathbf{w}_1^v = 0$
Pro i=1,2, ..., m

- Algoritmus obdrží akustický příznakový vektor $\bar{\mathbf{x}}_i$ odpovídající fonému y_i
- Predikce

$$\hat{y}_i = \arg\max_{v \in \mathcal{Y}} \sum_{u \in \mathcal{P}(v)} \text{mean}(\mathbf{w}_i^u \cdot \bar{\mathbf{x}})$$

- Trénovací algoritmus obdrží správný foném y_i
- V případě chybné predikce ($\gamma(\cdot,\cdot) \neq 0$) je vypočtena chybová funkce $\ell(\{\mathbf{w}_i^v\}, \mathbf{x}_i, y_i)$
- Znovu-výpočet váhovacích vektorů:

$$\mathbf{w}_{i+1}^v = \mathbf{w}_i^v + \alpha_i^v \cdot \text{mean}(\bar{\mathbf{x}})$$

$$\alpha_i^v = \begin{cases} \alpha_i & v \in \mathcal{P}(y_i) \backslash \mathcal{P}(\widehat{y}_i) \\ -\alpha_i & v \in \mathcal{P}(\widehat{y}_i) \backslash \mathcal{P}(y_i) \\ 0 & \text{jinak} \end{cases}$$

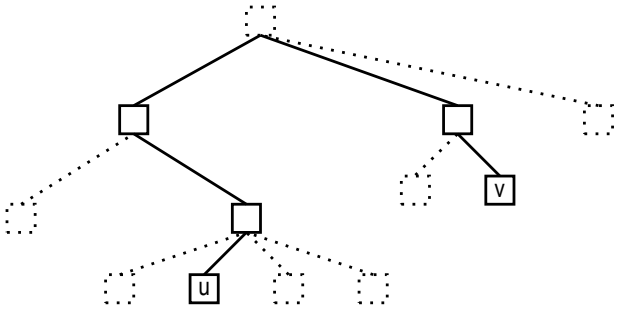
kde

$$\alpha_i = \frac{\ell(\{\mathbf{w}_i^v\}, \mathbf{x}_i, y_i)}{\gamma(y_i, \widehat{y}_i) \cdot \|\mathbf{x}\|_N}$$

Obr. 3.2: Navržený iterační trénovací algoritmus pro lineární klasifikátor s využitím hierarchické struktury.

Připomeňme, že pouze ta množina váhovacích vektorů $\{\mathbf{w}^v\}$ definovaná cestou $\mathcal{P}(v)$, je v každém kroku modifikována [8]. Z obrázku 3.3 je zřejmé, že pouze vrcholy vyznačené tučnou čarou jsou znovu odvozeny a zbytek váhovacích vektorů zůstává beze změn.

Analytické řešení optimalizační rovnice (3.10) je obtížné určit přímo, a proto je pro výpočet využito teorie tzv. Lagrangeových multiplikátorů (někdy také nazývané Lagrangeovy neurčité koeficienty) [3, 15]. V praxi je ovšem obvykle požadovaný přímý numerický výpočet, který se v případě SVM většinou spoléhá na Platův algoritmus SMO (Sequence Minimal Optimalization) [21]. Po náročném matematickém odvození dostáváme následující rovnice pro výpočet váhovacích vektorů $\{\mathbf{w}\}$:



Obr. 3.3: Demonstrace pro znovu-výpočet váhovacího vektoru – pouze tučně označené vrcholy v hierarchické struktuře jsou upraveny.

$$\mathbf{w}_{i+1}^{v} = \mathbf{w}_{i}^{v} + \alpha_i \text{mean}(\mathbf{x}), \quad v \in \mathcal{P}(y_i) \backslash \mathcal{P}(\widehat{y}_i) \tag{3.11}$$

$$\mathbf{w}_{i+1}^{v} = \mathbf{w}_{i}^{v} - \alpha_i \text{mean}(\mathbf{x}), \quad v \in \mathcal{P}(\widehat{y}_i) \backslash \mathcal{P}(y_i), \tag{3.12}$$

kde výpočet Lagrangeových multiplikátorů α_i je proveden na základě následujícího vzorce.

$$\alpha_i = \frac{\ell(\{\mathbf{w}_i^v\}, \mathbf{x}_i, y_i)}{\gamma(y_i, \widehat{y}_i) \cdot \|\mathbf{x}\|_N}, \tag{3.13}$$

kde $\|\cdot\|_N$ reprezentuje maticovou normu definovanou podle následujícího výrazu

$$\sum_{i=1}^{\max(i)} \sqrt{\sum_{j=1}^{\max(j)} x_{i,j}^2}, \tag{3.14}$$

a mean($\mathbf{x}$) je průměrná hodnota pro odpovídající příznakový vektor $\mathbf{x}$. Můj navržený algoritmus může být dále doplněn o nelineární transformaci příznakového prostoru $\mathcal{X}$ [20] implementací tzv. jádrových (kernel) funkcí $K(\cdot, \cdot)$. Pro více informací doporučuji literaturu [8, 3].

Algoritmus v každém kroku odvozuje nové váhovací vektory $\mathbf{w}_i$ tak, že pro fonémy a fonémové skupiny obsažené v cestě $\mathcal{P}(v)$ určí nové váhovací vektory a zbytek váhovacích vektorů doplní o odpovídající předchůdce. V případě, že vstupní databáze obsahuje m trénovacích promluv, je k dispozici m dílčích váhovacích vektorů pro každý foném a fonémovou skupinu $\{\mathbf{w}^v\}_{i=1}^m$. Podle původních předpokladů by měl poslední váhovací vektor dosahovat nejlepších výsledků, ovšem v praxi se ukazuje, že tomu tak není, a lepší výsledky jsou dosaženy vytvořením průměrného váhovacího vektoru ze všech m dílčích váhovacích vektorů $\{\mathbf{w}_i^v\}_{i=1}^m$ [20].

$$\mathbf{w}_{avg}^{v} = \frac{1}{m+1} \sum_{i=1}^{m+1} \mathbf{w}_i^v. \tag{3.15}$$

4 NELINEÁRNÍ KLASIFIKÁTORY

4.1 Nelineární hierarchické klasifikátory

Jak již bylo poznamenáno v předchozí kapitole, klasifikační funkce lze dále modifikovat tak, aby umožňovala pracovat s transformovanými příznaky. Hlavní myšlenkou pro transformaci příznaků je lepší separace vstupních dat. Z teorie SVM a jádrových metod definujeme tzv. Gramovu matici určující jádro této nelineární operace.

Nelineární hierarchické klasifikátory principiálně vychází z lineárních hierarchických klasifikátorů definovaných v kapitole 3.1. Nelinearita klasifikátoru je určena tvarem klasifikační funkce, která obsahuje nelineární operaci – tzv. jádrovou funkci $K(\mathbf{x}_i \cdot \mathbf{x}_j)$. V každém iteračním kroku jsou všechny odpovídající váhy modifikovány (přičtením nebo odečtením) o násobek $\alpha_i^u \cdot \mathbf{x}_i$, kde α_i^u je i-tý Lagrangeův koeficient a $\mathbf{x}_i$ je i-tý příznakový vektor. Výsledný váhovací vektor je pro m trénovacích vzorků určen rovnicí (4.1).

$$\mathbf{w}^u = \sum_{i=1}^{m} \alpha_i^u \cdot \mathbf{x}_i, \tag{4.1}$$

kde Lagrangeův koeficient α_i^u je určen podle vztahu (4.2).

$$\alpha_i^v = \begin{cases} \alpha_i & v \in \mathcal{P}(y_i) \backslash \mathcal{P}(\widehat{y}_i) \\ -\alpha_i & v \in \mathcal{P}(\widehat{y}_i) \backslash \mathcal{P}(y_i) \\ 0 & \text{jinak} \end{cases} \tag{4.2}$$

Výrazy $mathcal P(y_i) \backslash \mathcal{P}(\widehat{y}_i)$ a $\mathcal{P}(\widehat{y}_i) \backslash \mathcal{P}(y_i)$ definují množinu vrcholů na společné cestě mezi dvěma fonémy. Následnou substitucí rovnice (4.1) do (3.6) dostáváme následující rovnici definujicí nelineární klasifikátor:

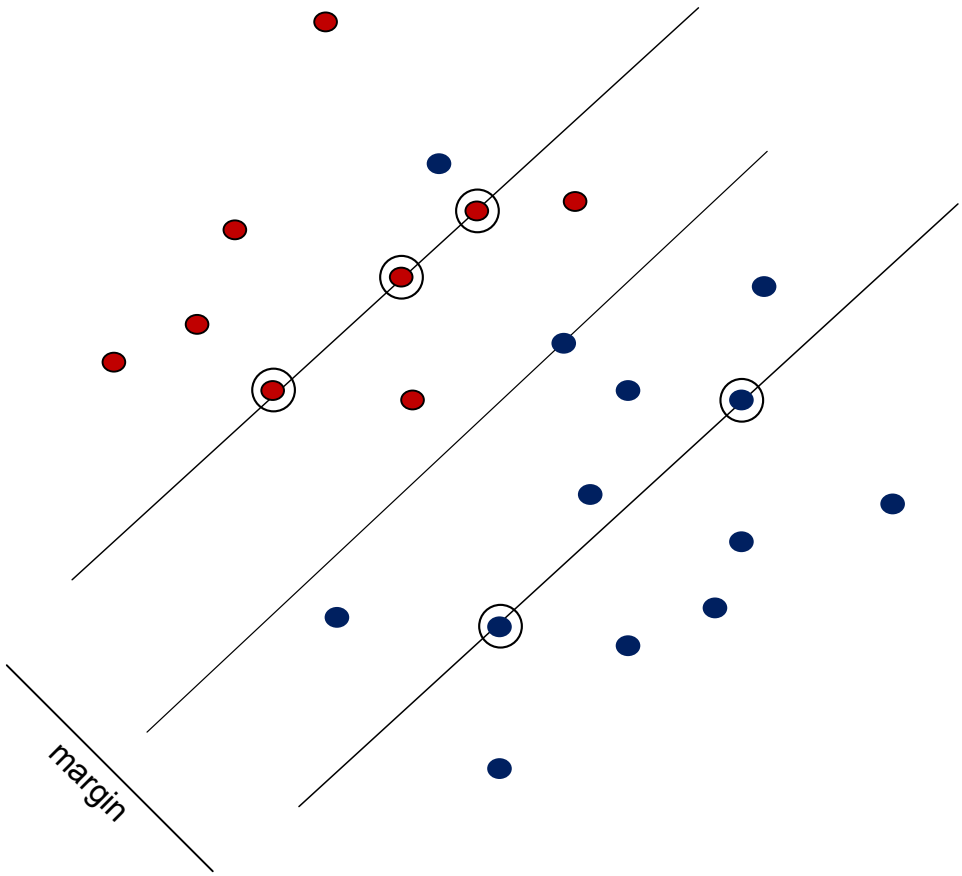
$$f(\mathbf{x}) = \arg\max_{v \in \mathcal{Y}} \sum_{u \in \mathcal{P}(v)} \sum_{i=1}^{m} \alpha_i^u \cdot \mathbf{x}_i \cdot \mathbf{x}. \tag{4.3}$$

Skalární součin $\mathbf{x}_i \cdot \mathbf{x}$ lze dále přepsat do podoby jádrové funkce $K(\mathbf{x}_i \cdot \mathbf{x}_j)$ a následnou substitucí dostáváme vztah pro nelineární hierarchickou klasifikační funkci $f(\mathbf{x})$

$$f(\mathbf{x}) = \arg\max_{v \in \mathcal{Y}} \sum_{u \in \mathcal{P}(v)} \sum_{i=1}^{m} \alpha_i^u \cdot K(\mathbf{x}_i, \mathbf{x}). \tag{4.4}$$

Z rovnice (4.4) je vidět, že pro výpočet klasifikační funkce $f(\mathbf{x})$ je zapotřebí celá vstupní trénovací databáze, což v případě rozsáhlé trénovací databáze může značně ovlivnit čas výpočtu. Jednou z možností, jak snížít výpočetní náročnost, je redukce množiny neurčitých koeficientů (a jim odpovídajících trénovacích vzorů) $\{\alpha_i^u; \mathbf{x}_i\}_{i=1}^m$ pouze na $\alpha_i^u > 0$. Z rovnic (4.1) a (4.2) je vidět, že tato operace neovlivní výsledek

výpočtu a z principu teorie SVM odpovídají nenulové neurčité koeficienty α_i bodům na rozhodovací nad-rovině [3].



Obr. 4.1: Princip metody podpůrných vektorů – koeficienty $\alpha_i^v > 0$ leží na rozhodovací nad-rovině

Ještě je nutné podotknout, že jádrová funkce $K(x_i, x_j)$ nemůže být definována libovolně a musí splňovat tzv. Mercelovy podmínky [3]. Mezi často používané jádrové funkce patří:

- Jádro s radiální bází
- Gausovo jádro
- Lineární jádro

4.2 Návrh efektivního trénovacího algoritmu pro nelineární hierarchický klasifikátor

Můj navržený trénovací algoritmus principiálně vychází z klasifikační funkce definované rovnicí (3.7). Implementace lineární klasifikační funkce do nelineárního

klasifikátoru byla zavedena na základě apriori znalosti obou klasifikátorů a hlavním důvodem pro zavedení je významné snížení složitosti výpočtu za cenu minimálního snížení úspěšnosti klasifikátoru. Nelinearita klasifikátoru je odvozena tvarem ztrátové funkce ℓ a vychází z tvaru ztrátové funkce definované rovnicí (3.9). Substitucí rovnice (4.1) do rovnice (3.9) a následnou modifikací sumy pro všechny vzorky do aktuálního kroku iterace dostáváme navrhovanou nelineární ztrátovou funkci pro hierarchický klasifikátor.

$$\ell = \left[\sum_{v \in \mathcal{P}(\widehat{y}_i)} \sum_{j<i} \|\alpha_j^v \cdot K(\mathbf{x}_i, \mathbf{x}_j)\| - \sum_{v \in \mathcal{P}(y_i)} \sum_{j<i} \|\alpha_j^v \cdot K(\mathbf{x}_i, \mathbf{x}_j)\| + \gamma(y_i, \widehat{y}_i)) \right]_+ \tag{4.5}$$

Funkce $[z]_+ = max\{z, 0\}$ a výraz $j < i$ reprezentuje součet pro všechny trénovací vzorky do aktuálního – definovaného aktuálním krokem iterace i. Nový váhovací vektor $\mathbf{w}_i^v$ je určen rovnicí (4.6)

$$\mathbf{w}_{i+1}^v = \mathbf{w}_i^v + \alpha_i^v \text{mean}(\mathbf{x}) \tag{4.6}$$

kde α_i^v odpovídá jednomu ze tří možných stavů, definovaných rovnicí (4.2) a α_i je pak určeno následujícím vztahem:

$$\alpha_i = \frac{\ell\left(\{\alpha_j^v\}_{j=1}^i, \mathbf{G}(j,i), y_i\right)}{\gamma(y_i, \widehat{y}_i) \cdot \mathbf{G}(i,i)}. \tag{4.7}$$

Výraz $\mathbf{G}(j,i)$, odpovídající hodnotě jádrové funkce $K(x_j, x_i)$, je součástí tzv. Gramovy matice $\mathbf{G}$. Gramova matice $\mathbf{G}$ obsahuje hodnoty všech dílčích jádrových výpočtů a slouží jako další optimalizační prvek – jádrové funkce jsou před samotnou klasifikací předvypočítány a dále pracujeme jen s maticí jednotlivých hodnot.

INICIALIZACE: $\forall v \in \mathcal{Y} : \mathbf{w}_1^v = 0, \alpha_i^v = 0$
Před výpočet Gramovy matice $\mathbf{G}$ [volitelné]

$$\mathbf{G}(i,j) = K(\mathbf{x}_i, \mathbf{x}_j)$$

Pro i=1,2, ..., m

- Algoritmus obdrží akustický příznakový vektor $\bar{\mathbf{x}}_i$ odpovídající fonému y_i
- Predikce

$$\hat{y}_i = \arg\max_{v \in \mathcal{Y}} \sum_{u \in \mathcal{P}(v)} \text{mean}(\mathbf{w}_i^u \cdot \bar{\mathbf{x}}_i)$$

- Trénovací algoritmus obdrží správný foném y_i
- V případě chybné predikce ($\gamma(\cdot,\cdot) \neq 0$) je vypočtena chybová funkce $\ell\left(\{\alpha_j^v\}_{j=1}^i, \mathbf{G}(j,i), y_i\right)$
- Znovu-výpočet váhovacích vektorů:

$$\mathbf{w}_{i+1}^v = \mathbf{w}_i^v + \alpha_i^v \cdot \text{mean}(\bar{\mathbf{x}})$$

$$\alpha_i^v = \begin{cases} \alpha_i & v \in \mathcal{P}(y_i) \backslash \mathcal{P}(\widehat{y}_i) \\ -\alpha_i & v \in \mathcal{P}(\widehat{y}_i) \backslash \mathcal{P}(y_i) \\ 0 & \text{jinak} \end{cases}$$

kde

$$\alpha_i = \frac{\ell\left(\{\alpha_j^v\}_{j=1}^i, \mathbf{G}(j,i), y_i\right)}{\gamma(y_i, \widehat{y}_i) \cdot \mathbf{G}(i,i)},$$

Obr. 4.2: Efektivní trénovací algoritmus pro nelineární hierarchická klasifikátor

5 GMM KLASIFIKÁTORY

5.1 GMM klasifikace

Klasifikátor založený na GMM modelování patří mezi tzv. generalizované metody (tzn. na základě vstupních trénovacích dat jsou odhadnuty parametry modelu, které pak popisují všechna další data) a vychází z teorie matematické pravděpodobnosti a statistiky [3, 22].

Princip GMM modelování je založen na lineární kombinaci dílčích Gaussových funkcí, kde každá funkce je definována svou střední hodnotou μ a směrodatnou odchylkou σ. Pro každou dílčí funkci je dále definována váha π. Vzhledem k tomu, že obvykle pracujeme s vícerozměrnými, vzájemně korelovanými daty, je směrodatná odchylka nahrazena kovarianční maticí $\mathbf{\Sigma}$ resp. v ideálním, nekorelovaném případě, diagonální kovarianční maticí [5]. Matematický zápis vícerozměrné Gaussovy funkce je definován jako:

$$\mathscr{N}(\mathbf{x}|\boldsymbol{\mu}, \mathbf{\Sigma}) = \frac{1}{\sqrt{(2\pi)^P |\mathbf{\Sigma}|}} \cdot e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})\mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})} \tag{5.1}$$

kde P je dimenze vstupních dat (počet parametrů dílčího příznakového vektoru). Z Bayesovy teorie podmíněné pravděpodobnosti je možné zapsat GMM model jako

$$p(\mathbf{x}) = \sum_{k=1}^{K} p(k) \cdot p(\mathbf{x}|k). \tag{5.2}$$

Zde $p(k)$ reprezentuje apriori informaci (váhu k-té dílčí Gaussovy funkce) a $p(\mathbf{x}|k)$ představuje dílčí Gaussovu funkci $\mathscr{N}(\mathbf{x}, \boldsymbol{\mu}, \bar{\mathbf{\Sigma}})$. Po drobném přepisu a s využitím rovnice (5.1) je možné vyjádřit GMM model následující rovnicí

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \cdot \mathscr{N}(\mathbf{x}|\boldsymbol{\mu}_k, \bar{\mathbf{\Sigma}}_k) \tag{5.3}$$

a současně musí platit, že

$$\sum_{k=1}^{K} \pi_k = 1. \tag{5.4}$$

V případě, že je řečový signál zpracováván současně ve více „proudech“ (tzn. že stejný úsek signálu je popsán více příznakovými vektory) [22] lze rovnici (5.3) upravit do následujícího tvaru

$$p(\mathbf{x}) = \prod_{r=1}^{R} \left[\sum_{k=1}^{K} \pi_k \cdot \mathscr{N}(\mathbf{x}|\boldsymbol{\mu}_k, \bar{\mathbf{\Sigma}}_k) \right]^{g_r}. \tag{5.5}$$

Exponent g_r vyjadřuje váhu odpovídajícího r-tého datového proudu. Ve většině případů je obvykle využíván jen jeden datový proud – tzn. tedy platí $R = 1, g_r = 1$ a rovnice (5.5) přechází na tvar rovnice (5.3).

Klasifikační funkce pro navrhovaný GMM klasifikátor má pak následující tvar

$$f(\mathbf{x}) = \arg\max_{C} \sum_{k=1}^{K} \pi_k^C \cdot \mathscr{N}(\mathbf{x}|\boldsymbol{\mu}_k^C, \bar{\boldsymbol{\Sigma}}_k^C) \tag{5.6}$$

kde hledáme maximum ze všech C tříd. V praktických aplikacích jsou klasifikátory často doplněny o prahovací konstantu b, která určuje minimální potřebnou úroveň pro odpovídající detekci (klasifikaci).

5.1.1 Trénovací algoritmus

Trénovací algoritmus vychází z kritéria maximální věrohodnosti ML (Maximum Likelihood) a pro maximalizaci věrohodnostní funkce se obvykle využívá iterativní Baum-Welschův BW (resp. EM) algoritmus [22]. Kritérium maximální věrohodnosti předpokládá, že existuje model $\mathrm{P}(\bar{\mathbf{x}}|\Theta)$ s neznámými parametry (pro GMM klasifikátor jsou to střední hodnota μ a kovarianční matice Σ a váha dílčí složky π) a cílem trénovacího algoritmu je odvození těchto parametrů na základě trénovacích dat $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m$. Obvykle je požadováno natrénovat více odlišných modelů a trénovací databáze pak tvoří množinu C dílčích trénovacích dat ve tvaru $\mathcal{S} = \{\bar{\mathbf{x}}_c\}_{c=1}^{C}$, kde C je celkový počet modelů (v praxi např. počet trénovaných fonémů). Princip metody ML využívá tzv. Fisherovu věrohodnostní funkci $\mathrm{F}(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N|\Theta)$ [22], která je definovaná jako:

$$\mathrm{F}(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N|\Theta) = \prod_{n=1}^{N} \mathrm{P}(\mathbf{x}_n|\Theta). \tag{5.7}$$

Cílem je nalézt maximum této funkce vzhledem k neznámým parametrům Θ

$$\hat{\Theta} = \arg\max_{\Theta} \prod_{n=1}^{N} \mathrm{P}(\mathbf{x}_n|\Theta). \tag{5.8}$$

V praxi se ovšem z technických důvodů většinou pracuje s logaritmem věrohodnostní funkce $\mathrm{F}(\bar{\mathbf{x}}|\Theta)$ a rovnice (5.8) má pak následující tvar

$$\hat{\Theta} = \arg\max_{\Theta} \sum_{n=1}^{N} \log \mathrm{P}(\mathbf{x}_n|\Theta). \tag{5.9}$$

Maximalizace definovaná rovnicí (5.8), resp. rovnicí (5.9) je komplikovaná úloha, která nemá explicitní řešení [22]. Existuje ovšem iterační algoritmus EM, který zaručuje v každém kroku zlepšení stávajících algoritmů a tedy konvergenci k maximu

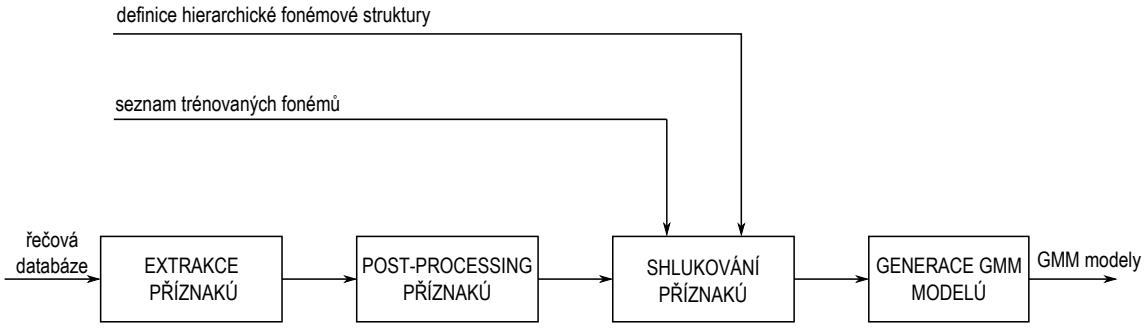
věrohodnostní funkce (algoritmus je detailně popsán např. v literatuře [22, 3]). Algoritmus bohužel nezaručuje konvergenci ke globálnímu maximu a často je dosaženo pouze lokálního maxima [14]. Jednou z možností je vícenásobná počáteční inicializace EM algoritmu a následná verifikace např. na kros-validační množině [18].

5.2 Návrh GMM klasifikátoru s implementací hierarchické struktury

GMM klasifikátor definovaný rovnicí 5.6 nezohledňuje hierarchickou fonémovou strukturu. Implementace hierarchické struktury do GMM klasifikátoru je provedeno tak, že při klasifikaci je brán ohled na výstupní hodnotu rodičovské třídy $\mathcal{A}^1(v)$ fonému v. To znamená, že v trénovacím procesu musí být také trénovány dílčí fonémové třídy. Trénovací data pro tyto dílčí fonémové třídy jsou získána jednoduchým sloučením všech trénovacích dat fonémů v_i patřící do dané fonémové třídy $\mathcal{A}^1(v)$. Moje navržená klasifikační funkce má pak následující tvar

$$f(\mathbf{x}) = \arg\max_{C} \left[p_C(\mathbf{x}) + p_C^{\mathcal{A}^i(C)}(\mathbf{x}) \right], \tag{5.10}$$

kde $p_C^{\mathcal{A}^i(C)}(\mathbf{x})$ je hodnota pro rodičovskou třídu C a funkce $p(\mathbf{x})$ se určí z rovnice (5.3). Z důvodu dodržení obecné formulace klasifikátoru je zde využito proměnné C definující danou třídu. V kontextu hierarchické fonémové struktury je běžné využívat proměnnou v pro danou třídu. Obě proměnné mají zde v textu ovšem stejný význam.



Obr. 5.1: Blokové schéma GMM trénovacího algoritmu s implementací hierarchické struktury

Trénovací algoritmus je znázorněn na obrázku 5.1 a 5.2. Na vstupu systému je řečová databáze, seznam fonémů, které chceme natrénovat a definice hierarchické fonémové struktury. Příklad takové struktury je znázorněn na obrázku 3.1 a **??**.

Navzorkovaný řečový signál je převeden na posloupnost příznakových vektorů $\bar{\mathbf{x}}$ a případně ještě modifikován v bloku post-processingu (např. redukce dimenze pomocí metody hlavních komponent - PCA (Principal Component Analysis)). Následuje zařazení jednotlivých fonémů do dílčích tříd a vytvoření rodičovských tříd $\mathcal{A}^1(v)$ sloučením všech odpovídajících příznaků. Výstupem trénovacího algoritmu jsou pak GMM modely pro jednotlivé fonémy a fonémové třídy.

Z rovnice (5.10) je vidět, že výsledná hodnota klasifikátoru je ovlivněna pouze rodičovskou třídou první úrovně (na rozdíl od klasifikátorů definovaných v předchozí kapitole).

INICIALIZACE: definice vstupních parametrů modelu

- počet GMM komponent.
- kritérium optimality.
- Definice fonémových skupin na základě hierarchické struktury.

Pro $\forall v \in \mathcal{Y}$

- Algoritmus obdrží trénovací množinu pro každý foném v.
- Pro každou definovanou fonémovou skupinu vytvořím množinu trénovacích dat (shlukováním dat)
- Inicializace EM algoritmu
- Re-estimace algoritmu dokud není dosaženo optimálního kritéria.

Klasifikační funkce

$$f(\mathbf{x}) = \arg\max_{C} \left[p_C(\mathbf{x}) + p_C^{\mathcal{A}^i(C)}(\mathbf{x}) \right],$$

Obr. 5.2: Efektivní trénovací algoritmus pro GMM klasifikátor s implementaci hierarchické struktury

6 ZÁVĚR

Tato práce se zabývala návrhem systému pro detekci klíčových slov a jeho následnou optimalizací. Stěžejní částí detekčního systému je fonémový klasifikátor, a proto byl hlavní důraz této práce kladen hlavně na fonémovou klasifikaci a současně na optimalizaci trénovacích algoritmů.

V práci byly prezentovány dva odlišné přístupy pro rámcovou fonémovou klasifikaci. První přístup byl založen na GMM modelování dílčích fonémů a druhý na konstrukci separační nadroviny. Všechny prezentované klasifikátory byly dále doplněny hierarchickou strukturu pro daný jazyk. V prací byly také navrženy dva efektivní trénovací algoritmy a byla provedena implementace hierarchické struktury do GMM klasifikátoru.

V kapitole 3.1: *Lineární hierarchické klasifikátory* byl definován lineární rámcový klasifikátor založený na implementaci hierarchické struktury. Klasifikátor principiálně vychází z návrhu podle předlohy O. Dekela a dosahuje přibližně srovnatelných výsledků v porovnání se standardním GMM rámcovým klasifikátorem. Hlavní předností algoritmu je zaručená konvergence trénovacího algoritmu ke globálnímu maximu, na rozdíl od GMM klasifikátoru. Hlavní nevýhodou je mnohem vyšší čas potřebný pro natrénování klasifikátoru. Algoritmus popsaný v kapitole 3.2: *Návrh efektivního trénovacího algoritmu pro lineární hierarchický klasifikátor* implementuje efektivní trénovací algoritmus pro lineární hierarchický klasifikátor.

V kapitole 4.1: *Nelineární hierarchické klasifikátory* byla popsána implementace nelineárních jádrových funkcí do stávajících lineárních hierarchických rámcových klasifikátorů. Z naměřených výsledků je vidět, že nelineární klasifikátory dosahují o něco lepší výsledky (cca o 2 procenta), ale s mnohem vyššími výpočetními nároky pro natrénování klasifikátoru (více než 2x). Na základě předchozích pozitivních výsledků s lineárními klasifikátory byl navržen efektivní trénovací algoritmus pro nelineární hierarchické rámcové klasifikátory (viz kapitola 4.2: *Návrh efektivního trénovacího algoritmu pro nelineární hierarchický klasifikátor*). Výsledky opět ukazují, že implementace sekvenčního trénovacího algoritmu zásadně snížila čas potřebný pro natrénování klasifikátoru (více než 2x). Nelineární klasifikátor implementující navržený trénovací algoritmus dosahoval vůbec nejlepších výsledků jak na metrikách PER a MISS, tak současně na metrice AUC.

Klasifikátor založený na GMM modelování nezahrnuje hierarchickou strukturu. V kapitole 5.2: *Návrh GMM klasifikátoru s implementací hierarchické struktury* byl proto navržen GMM klasifikátor, který implementuje hierarchickou strukturu.

LITERATURA

[1] at all., S. Y.: *The HTK Book*. Cambridge University Engineering Department, třetí vydání, december 2005.
URL `http://htk.eng.cam.ac.uk/docs/docs.shtml`

[2] Aradilla, G.; Vepa, J.; Bourlard, H.: Using posterior-based features in template matching for speech recognition. Technická Zpráva IDIAP-PR 06-23, IDIAP research institute, June 2006.

[3] Bishop, C. M.: *Pattern recognition and Machine learning*, ročník 3. Springer, February 2006, ISBN 0-387-0387-31073-8.

[4] Burget, L.: *Complementarity of Speech Recognition Systems and System Combination*. Dizertační práce, Brno University of Technology, September 2004.

[5] Cernocky, J.: Zpracování řečových signálů. Prosinec 2006.

[6] Chopra, S.; Hadsell, R.; LeCun, Y.: Learning a Similarity Metric Discriminatively with Application to Face Verification. In *Proceedings of the conference CVPR2005*, IEEE Computer Society, 2005, ISBN 0-7695-2372-2, str. 8.
URL `http://yann.lecun.com/exdb/publis/pdf/chopra-05.pdf`

[7] Crammer, K.; Dekel, O.; Shalev-Shwartz, S.; aj.: Online passive-aggressive algorithms. In *Advances in Neural Information Processing Systems 16*, Cambridge: MIT Press, 2004.

[8] Dekel, O.; Keshet, J.; Singeer, Y.: An Online Algorithm for Hierarchical Phoneme Classification. *Springer*, ročník Volume 3361/2005, January 2005: s. 146–158.

[9] Ghoshal, A.; Povey, D.; Agarwal, M.; aj.: A novel estimation of feature-space MLLR for full-covariance models. In *Proc. International Conference on Acoustics, Speech, and Signal Processing*, ročník 2010, IEEE Signal Processing Society, 2010, ISBN 978-1-4244-4296-6, ISSN 1520-6149, s. 4310–4313.
URL `http://www.fit.vutbr.cz/research/view_pub.php?id=9308`

[10] Grangier, D.; Bengio, S.: Learning the inter-frame distances for Discriminative Template-based Keyword Detection. In *Proceedings of the conference INTERSPEECH*, 2007, str. 4.
URL `http://david.grangier.info/`

[11] Grangier, D.; Keshet, J.; Bengio, S.: *Discriminative keyword Spotting*, kapitola 11. Wiley, January 2009, s. 115–137.

[12] Grézl, F.: *Trap-based Probabilistic features for Automatic Speech Recognition.* Dizertační práce, Brno University of Technology, September 2007.

[13] Juravsky, D.; Martin, J. H.: *Speech and language processing.* Upper Saddle River, New Jersey 07458: Prentice Hall, druhé vydání, 2008, ISBN 978-0-13-187321-6.

[14] Keshet, J.: *Large Margin Algorithms for Discriminative Continuous Speech Recognition.* Dizertační práce, Hebrew University, September 2007.

[15] Keshet, J.; Bengio, S. (editoři): *Automatic Speech and Speaker Recognition: Large Margin and Kernel Methods*, ročník 1. Wiley, January 2009, ISBN 978-0-470-69683-5, 268 s.

[16] Keshet, J.; Grangier, D.; Bengio, S.: Discriminative Keyword Spotting. In *Workshop on Non-Linear Speech Processing (NOLISP)*, 2007, str. 5.
URL `http://david.grangier.info/pub/papers/2007/KeshetGrBe07.pdf`

[17] Keshet, J.; Grangier, D.; Bengio, S.: Discriminative Keyword Spotting. *Speech Communication*, ročník 51, April 2009: s. 317–329, ISSN 0167-6393.
URL `http://dx.doi.org/10.1016/j.specom.2008.10.002`

[18] Matton, M.; Watchter, M. D.; Compernolle, D. V.; aj.: Maximum Mutual Information Training of Distance Measures for Template Based Speech Recognition. In *10th International Conference on Speech and Computer*, SPECOM 2005 Proceedings vol:2, 2005, ISBN 5-7452-0110-x, str. 4.

[19] Pfeifer, V.; Balik, M.: Comparison of current frame-based phoneme classifiers. *Advances in Electrical and Electronic Engineering*, ročník 9, č. 5, December 2011: s. 243–250.
URL `http://advances.utc.sk/index.php/AEEE/article/view/545`

[20] Pfeifer, V.; Balik, M.; Malý, J.: Frame based phoneme classification using large margin and kernel methods. In *Telecomunications and Signal Processing – TSP 2010*, Baden, Austria, September 2010, str. 4.

[21] Platt, J. C.: Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. Technická zpráva, ADVANCES IN KERNEL METHODS - SUPPORT VECTOR LEARNING, 1998.

[22] Psutka, J.; Muller, L.; Matoušek, J.; aj.: *Mluvíme s počítačem česky.* ACADEMIA, October 2006, ISBN 80-200-1309-1.

[23] Szoke, I.: *Keyword Detection in speech data*. Dizertační práce, Brno University of Technology, 2009.

Ing. Václav Pfeifer

CONTACT INFORMATION

Mladotická 758
Slavičín
763 21 Czech Republic

Phone: +420 608 312 898
E-Mail: vaclav.pfeifer@gmail.com

OBJECTIVE

Software Architect, project manager, team leader.

EDUCATION

Brno University of Technology, Brno, Czech Republic

Doctoral's degree, Faculty of Electrical Engineering, Department of Telecommunivations
09/2006 - 01/2010 (thesis in review)

- Specialization: Classification and Machine Learning. Signal and speech processing.
- Thesis topic: *Keyword detection in spoken data*
  Design of a new keyword detection system based on the non-linear feature functions. Proof of concept designed and implemented in MATLAB.

Master's degree, Faculty of Electrical Engineering,
09/2001 - 01/2006

- Specialization: Software Engineering, communication technologies
- Thesis topic: *Arithmetic precision library*
  Design and implementation of a library in C++ for matrix operations in arbitrary precision.

PROFESSIONAL EXPERIENCE

Honeywell, Brno, Czech Republic

Scientist, Software Architect, Team leader **12/2010 till now**

- Focal for speech processing, state estimation filters and mathematic optimization.
- Responsibility for team/company representation by participation on multiple technology symposiums.
- Responsibility for a software architecture on multiple projects.
- Communication with the customer.
- Supervision on multiple projects.
- Leading and coaching.
- Deputy of manager.

Faculty of Electrical Engineering, Brno, Czech Republic

Project manager, Software Engineer (embedded) **06/2007 - 09/2011**

- Preparation of learning materials.
- Software design of multiple embedded applications.
- Project management.
- Research of speech and signal processing.
- Presentation of the results.
- Communication with the customer.

Prototypa a.s. Brno, Czech Republic

Software Engineer **01/2005 09/2006**

- Design of a measurement system in LABVIEW.
- Responsibility for proper documentation.
- Communication with the customer.

Awards and contests

Multiple **Bravo Awards** for an excellence work on Honeywell projects.

Best of paper from **Research in Telecommunication and Technology (RTT)** 2010 international conference.

Projects

Air Traffic Capacity Simulator, *JAVA*, *MATLAB*
Design of software architecture in EA. Implementation in JAVA using repast simphony framework. Preparation of proper test-scenarios, output data post-processing and graphical representation in MATLAB. Weekly communication with US and China customer.

D3CoS, *C++*, *MATLAB*
Mathematic optimization of the aircraft vertical profile. Design of software GUI interface in MATLAB. Design and implementation of the core algorithm. Responsibility and supervision for multiple work packages.

SESAR 9.29, *C++*, *MATLAB*
Design of the core tracking algorithm and implementation in C++. Design and implementation of the fast-time simulation in C++.

SuperFlow, *C#*, *VB*, *SQL*
Design and implementation of a form application for data processing and evaluation. Design of SQL schema (MS SQL). Responsibility for PI/PM. Communication with external customer.

Personal, *JAVA*
Design of the web interface using JAVA (GWT, OSGI). Responsibility for software architecture.

Certificates and Courses

Green Belt certification - practical knowledge with 6-Sigma tools

Project Management (PI/PM 1+2)

Communication, **Presentation** and **Negotiation**

Computer skills

Programming languages JAVA, C/C++/C#, MATLAB, LabView, Bash, HTML

Databases MySQL, PostgreSQL, MsSQL

Frameworks OSGi, GWT, SmartGWT, Repast Simphony

Operating systems Microsoft Windows, GNU/Linux

Development tools Eclipse, NetBeans, EA, IBM Rhapsody, MS Visual Studio

Office tools MS Office, LaTeX

Other Design Patterns

Language Skills

English advanced

German basic knowledge of the language

Czech native speaker

Miscellaneous

driving license category B, clean driving record

Personal qualities

Drive, Motivation, Team player, flexible, hard worker

Hobbies

Muay Thai, athletics, swimming, chess, programming, astrophysics

ABSTRAKT

Systémy pro zpracování řečových signálů jsou vyvíjeny již delší dobu, ale až s nástupem výkonných výpočetních systémů se začalo s integrací těchto systémů do praxe. Tato disertační práce se zabývá návrhem systému pro detekci klíčových slov v řečových signálech. Navržený systém principiálně vychází z *Large Margin and Kernel* metod a klíčovou součástí systému je fonémový klasifikátor. Byly navrženy dva hierarchické klasifikátory – lineární a nelineární, spolu s efektivním trénovacím algoritmem. Současně byl navržen klasifikátor založený na „Gaussian Mixture Models“ s implementací hierarchické struktury. Důležitou součástí detekčního systému je extrakce příznaků, a proto byl navržený systém vyhodnocen na současně nejrozšířenějších extrakčních metodách. Součástí technického řešení práce byla implementace detekčního systému v prostředí MATLABU a návrh hierarchické fonémové struktury pro Český jazyk. Všechny algoritmy byly vyhodnoceny pro Český a Anglický jazyk na databázích (DBRS a TIMIT)

KLÍČOVÁ SLOVA

klasifikátor, rámcový, foném, detekční, hierarchický, řeč

ABSTRACT

Speech processing systems have been developed for many years but the integration into devices had started with the deployment of the modern powerful computational systems. This dissertation thesis deals with development of the keyword detection system in speech data. The proposed detection system is based on the Large Margin and Kernel methods and the key part of the system is phoneme classifier. Two hierarchical frame-based classifiers have been proposed – linear and non-linear. An efficient training algorithm for each of the proposed classifier have been introduced. Simultaneously, classifier based on the Gaussian Mixture Models with the implementation of the hierarchical structure have been proposed. An important part of the detection system is feature extraction and therefor all algorithms were evaluated on the current most common feature techniques. A part of the thesis technical solution was implementation of the keyword detection system in MATLAB and design of the hierarchical phoneme structure for Czech language. All of the proposed algorithms were evaluated for Czech and English language over the DBRS and TIMIT speech corpus.

KEYWORDS

classifier, frame-based, phoneme, detection, hierarchical, speech

PFEIFER, Václav *Detekce klíčových slov v řečových signálech*: dizertační práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2012. 27 s. Vedoucí práce byl Ing. Miroslav Balík, Ph.D.